Named Entity Recognition in Italian Using CRF

Peng Cai[1], HangZai Luo[2], and AoYing Zhou[2]

Institute of Massive Computing
East China Normal University, 200062 Shanghai, China
[1] pengcaibox@tom.com
[2] {hzluo, ayzhou}@sei.ecnu.edu.cn

Abstract. We present a system for name entity recognition in Italian, which was implemented via CRF. Generally speaking, the initial output of CRF has good precision but bad recall. The problem was resolved by using state confidence to rectify the initial state. For example, the best label or state "O" can be replaced with the second best state i.e. "B" or "I" in IOB2 format using heuristic rules. Although this approach is simple, it can efficiently improve recall, which makes the overall F-measure increase obviously.

Keywords: named entity, conditional random field, state confidence.

1 Introduction

Named entity recognition is an important task for many applications, such as information extraction, information retrieval and question answer. In Evalita 2009, our system based on CRF (conditional random field) [2] only learned from training data without using any external resources. In our system, the component of CRF was based on open source software CRF++ [1].

2 The System

The implementation of our system contains two steps. Firstly we need to define feature templates for CRF++, and then to rectify the initial output with state confidence. In the following, we discuss them respectively.

2.1 Feature Templates

CRF transforms the problem of entity recognition to sequence label problem. Given a observation sequence X, which is a vector including n words, i.e. x_1, x_2, … x_n, Y is a label or state sequence corresponding to X containing n labels or states, i.e. y_1, y_2, … y_n. The state set contains three distinct states, i.e. "B", "I", and "O", which means the beginning, intermediate, and out of entity respectively. CRF can capture the

relationship between observation and state sequence via feature function f(X, Y). Generally, f is indicator function. To linear CRF, the form of feature function is f(X, y_{i-1}, y_i) which reflects the association between observation sequence X, previous and current state. In real application, we often take the subsequence of X as the first parameter; otherwise too many sparse feature functions will be produced.

In CRF++, feature functions are produced according to predefined feature templates, which defined in our system as follows:

Template_1: F(x_{i-2},y_i)
Template_2: F(x_{i-1},y_i)
Template_3: F(x_i, y_i)
Template_4: F(x_{i+1},y_i)
Template_5: F(x_{i+2},y_i)
Template_6: F(x_{i-1}, x_i,y_i)
Template_7: F(x_i,x_{i+1}, y_i)
Template_8: F(x_{i-2},x_{i-1},x_i,y_i)
Template_9: F(x_{i-1},x_i,x_{i+1}, y_i)
Template_10: F(x_i,x_{i+1},x_{i+2},y_i)
Template_11: F(y_{i-1},y_i)

In the above, x_i represents observation word or POS, and y_i denotes state respectively. The subscript denotes the relative position. Templates, from 1 to 10, defined feature function set between current state and observation subsequence, where the size of window for X is five. Template 11 defined the state transformation function between neighbor states.

2.2 State Confidence

Generally speaking, the best state sequence, i.e. Y= argmax P(Y/X), was taken as the output of system. However, we found that this may lead to perfect precision but bad recall. Moreover, the second best state sequence may also not the true good output.

According to CRF, given a state sequence Y, we can calculate P(Y/X) via CRF. Similarly, given y_i, P(y_i/X) can be used to measure accuracy of single state output. We call P(y_i/X) as state confidence. It is observed that if single state output, i.e. y_i, have the value "O" and P(y_i="B"/X) or P(y_i="I"/X) exceed predefined threshold, we can use "B" or "I" to replace "O" for improving recall while keeping precision in a reasonable range. State confidence is defined as follows:

$$P(y_i = \text{"B"}/X) = \sum_{\substack{Y' = y_1, y_2 \ldots y_i \ldots y_n \\ \text{AND } y_i = \text{"B"}}} P(Y'/X) \qquad (1)$$

P(y_i="I"/X) can be defined similarly, all of which can be calculated effectively by dynamic programming [3].

We rectify each state y_i using a heuristic rule, which defined as follows:

$$y_{update\text{-}i}=\begin{cases}\text{"B"} & P(y_i=\text{"B"}/X)>\theta \text{ and } P(y_i=\text{"B"}/X)>P(y_i=\text{"I"}/X)\\ \text{"I"} & P(y_i=\text{"I"}/X)>\theta \text{ and } P(y_i=\text{"I"}/X)>P(y_i=\text{"B"}/X)\\ \text{"O"} & P(y_i=\text{"B"}/X)<\theta \text{ and } P(y_i=\text{"I"}/X)<\theta\end{cases} \quad (2)$$

Where if the initial state of y_i is "O" we can modify it based on a predefined threshold θ and $y_{update\text{-}i}$ is the corresponding state to y_i after modification.

3 Experimental Results and Discussion

In the following, we present experimental results, including initial output of CRF and results after modification with different threshold θ. From Table 1, it is clear that imbalance exists in precision and recall. Table 2 to 5 present the results after initial output were rectified by θ. When θ=0.2, F1 can achieve the best. However, how to select θ depend on the specific requirements of real application. In Evalita 2009, we select θ=0.15 with cross validation and table 6 give the submitted result.

Table 1. Initial output of CRF for Evalita 2009 test set

Category	Precision	Recall	F1
Overall	74.16	48.83	58.89
GPE	77.31	55.73	64.77
LOC	95.00	12.18	21.59
ORG	68.32	37.32	48.27
PER	74.80	54.16	62.83

Table 2. Results for Evalita 2009 test set (θ=0.4)

Category	Precision	Recall	F1
Overall	73.16	50.00	59.40
GPE	77.67	57.22	65.89
LOC	95.00	12.18	21.59
ORG	66.40	39.10	49.22
PER	73.66	54.92	62.92

Table 3. Results for Evalita 2009 test set (θ=0.3)

Category	Precision	Recall	F1
Overall	71.96	52.46	60.68
GPE	77.25	59.41	67.16
LOC	95.24	12.82	22.60
ORG	63.96	42.13	50.80
PER	72.85	57.32	64.16

Table 4. Results for Evalita 2009 test set (θ=0.2)

Category	Precision	Recall	F1
Overall	68.05	55.34	61.04
GPE	75.13	62.12	68.01
LOC	80.65	16.03	26.74
ORG	60.06	46.31	52.30
PER	68.47	59.55	63.70

Table 5. Results for Evalita 2009 test set (θ=0.1)

Category	Precision	Recall	F1
Overall	62.01	59.38	60.67
GPE	71.34	66.84	69.02
LOC	75.00	17.31	28.13
ORG	50.35	49.88	50.12
PER	63.87	63.71	63.79

Table 6. UniEastChina_Cai_NER results submitted for Evalita 2009 test set (θ=0.15)

Category	Precision	Recall	F1
Overall	65.55	57.09	61.03
GPE	74.20	64.92	69.25
LOC	84.38	17.31	28.72
ORG	55.91	47.71	51.49
PER	66.17	61.02	63.49

References

1. CRF++: Yet Another CRF toolkit, http://crfpp.sourceforge.net/
2. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the 18th International Conference on Machine Learning, pp. 282--289. Morgan Kaufmann, San Francisco (2001)
3. Culotta, A., McCallum, A.: Confidence estimation for information extraction. In: Proceedings of HLT-NAACL, pp. 109--112 (2004)